SUBSTITUTE SPECIFICATION

MARKED-UP VERSION

UNIVERSAL GAMING ENGINE

CROSS REFERENCE TO RELATED APPLICATIONS

The present application is a continuation in part and incorporates by reference in its entirety, U.S. Application No. 09/143,907, filed August 31, 1998 entitled "UNIVERSAL GAMING ENGINE", which was a divisional application of U.S. Application No. 08/959,575 filed October 28, 1997, which issued on August 7, 2001 as U.S. Patent 6,272,223, which was a divisional application of U.S. Application No. 08/358,242 filed December 19, 1994 which issued on January 13, 1998 as U.S. Patent 5,707,286. Each of the foregoing patents and applications are incorporated by reference in their entirety.

This application also is a continuation in part, and incorporates by reference, in its entirety, U.S. Application No. 09/698,507, filed October 26, 2000 entitled "CRYPTOGRAPHY AND CERTIFICATE AUTHORITIES IN GAMING MACHINES" which is based on, and claims priority to U.S. Provisional Application No.: 60/161,591, filed October 26, 1999, which is also incorporated by reference, in its entirety.

The present application incorporates by reference, in its entirety, U.S. Application No. 10/116,424 filed April 3, 2002 entitled "SECURED VIRTUAL NETWORK IN A GAMING ENVIRONMENT" which is a continuation in part of application No. 09/732,650 filed on December 7, 2000, which is also incorporated by reference, in its entirety.

FIELD OF THE INVENTION

[0001] The present invention relates, in general, to gaming machines, and, more particularly, to an electronic gaming engine supporting multiple games and multiple users.

The present invention also relates to an apparatus and method for encrypting communications on a network bus in a gaming system, and more particularly, to an apparatus and method where a certificate authority server manages keys used to secure communications on a network bus in a gaming system.

2. Statement of the Problem

[0002] Casino gaming has grown rapidly in the United States. Casino gaming is experiencing similar growth throughout the world. An important segment of this developing industry is electronic games. An electronic implementation of a game requires a method for interpreting human actions as they occur within the constraints of the rules as well as the ability to respond with chance events.

[0003] Microprocessors allow games that formerly relied on analog devices for generating chance events, such as dice, to be simulated digitally. Simulating a die roll with a computer would seem to be a contradiction because the microprocessor is the embodiment of logic and determinism. With care, however, it is possible to create deterministic algorithms that produce unpredictable, statistically random numbers.

[0004] Contemporary games consist of a framework of rules that define the options for how a user or random event generator may change the game state. Play begins with an initial state. Subsequent play consists of user initiated events that trigger the execution of one or more rules. A rule may proceed deterministically or non-deterministically.

[0005] Typical games consist of deterministic and non-deterministic rules. A game progresses by the interaction of these rules. There are two sources for non-determinism: player decisions and chance events. In the game of Poker, for example, deciding to replace three instead of two cards in a hand is a player decision that is limited, but not pre-determined, by rules. The rules limit the range of options the player has, but within that set of options the player is free to choose. An example of a chance event is the random set of cards received by the poker player. Shuffled cards do not produce a predictable hand.

[0006] Other examples that illustrate determinism and non-determinism in gaming are popular casino pastimes such as Blackjack, Keno, and Slot machines. The first Blackjack hand a player receives is two cards from a shuffled deck. The number of cards dealt is two, but the cards could be any from the deck. Keno is essentially a lottery. In Reno, a player attempts to guess twenty balls chosen from a basket of eighty balls. The rules dictate that to participate, a player must fill out a Keno ticket indicating the balls he believes will be chosen in the next round. The selection of balls, however, is a purely random event. Slot machines require the player to pull a handle for each round. Slot wheels stop at random positions.

[0007] The non-deterministic problem in most parlor games is random sampling without replacement: given a set of n elements, randomly choose m of them without replacement where m is less than or equal to n. Although sampling without replacement covers most popular games, it would be easy to conceive of games that required replacement. For example, consider a variant of Keno that replaces each chosen ball before selecting the next ball. Until now, no device is available that services the needs of multiple games by providing algorithms for sampling with and without replacement as well as others such as random permutation generation, sorting, and searching.

[0008] A casino player must know the likelihood of winning a jackpot is commensurate with the stated theoretical probabilities of the game. Moreover, the casino would like to payout as little as possible while maximizing the number of their game participants. Because each game sponsored by a casino has a built-in theoretical edge for the house, over time and with repeated play, the house will make money. In other words, the casino does not need to cheat the customer because it has a built-in edge. The customer, who is at a disadvantage in the long run, will want to know the game is fair in order to manage risk. It is a theoretical fact that bold wagering in Roulette increases a player's odds of winning. A player who cannot know the odds of winning cannot formulate a strategy.

[0009] Provided that the deterministic rules of a game are implemented correctly, it is essential that the chance events of a game are indeed random. An important subproblem for generating random events is uniform random number generation. If the underlying uniform random number generator does not generate statistically independent and uniform pseudo-random numbers, then either the house or customer will be at a disadvantage. A poorly designed system might favor the housed initially and over time turn to favor the player. Certainly the house would not want this situation because it makes revenue projection impossible. Any regulatory body would like to ensure that neither the house nor customer have an advantage beyond the stated theoretical probabilities of the game. In the context of fairly implemented rules, the only way for the house to increase its revenue is to increase the number of players participating in their games.

[0010] Typically, an engineer creating an electronic game generates a flow chart representing the rules and uses a random number generator in conjunction with combinatorial algorithms for generating chance events. Representing rules is one problem. Generating chance events to support those rules is another. Creating pseudo-random numbers is a subtle problem that requires mathematical skills distinct from other problems of gaming. In other words, a skilled game programmer may be unable to solve the problems of developing a proper random number generator. Even if given a quality random number generator, problems can occur in hardware implementations that render the generator predictable. One example is using the same seed, or initial state, for the generator at regular intervals and repeatedly generating a limited batch of numbers. Without attending to the theoretical aspects of a uniform random number generator, it is not possible to implement the rules of a game perfectly. The result is a game unfair to the house, players, or both. Hence, there is a need for a gaming system, apparatus, and method that separate the problem of implementing game rules from that of random event generation.

[0011] The need for such a device is also evident at the regulatory level. Gaming is a heavily regulated industry. States, tribes, and the federal government have gaming regulatory agencies at various levels to ensure fairness of the games. The gaming regulatory authority certifies that a particular implementation of a game reflects the underlying probabilities. Because electronic games are implemented in often difficult to understand software, the problem of verifying fairness of a game is challenging. Further, there is little uniformity in the implementation of fundamental components of various games. To determine fairness, the gaming authority subjects each game to a battery of tests. No set of statistical tests performed on a limited portion of the random number generator period can ensure that the generator will continue to perform fairly in the field. The process of testing is both expensive and of limited accuracy. Hence, a regulatory need exists for a uniform, standardized method of implementing games that reduce the need and extent of individual game testing while increasing the reliability of detecting and certifying game fairness.

3. Solution to the Problem

[0012] The Universal Gaming Engine (UGE) in accordance with the present invention is a gaming apparatus providing a consistent game development platform satisfying the needs of the gaming authority, house, player, and game developer. The UGE's parates the problems of developing game rules from the difficulty of producing chance events to support those rules. Functions that are common to a number of games are included in the gaming engine so that they need not be implemented separately for each game. By including basic functions shared by a number of games, hardware costs are greatly reduced as new games can be implemented merely by providing a new set of rules in the rules library and the basic hardware operating the game remains unchanged.

Conventional gaming machines include a processor, a rules library, a random number generator and an interactive display. In the casino, these conventional gaming devices are, typically, stand-alone type machines. Increasingly, the gaming machines in a casino are networked via a network bus to a gaming server. This networking is desired because it allows the casino to monitor wagering and other activities performed at each of the networked gaming machines. Since the monitoring of wagering and other activities performed at each of the networked gaming machines can include financial information, the casino desires that the communications over the network bus be secure.

In considering secure gaming communications, there are several important goals that should be addressed. The network bus should ensure privacy. Privacy, also termed confidentiality, is the condition where the information is kept secret from all but those authorized to access the information. In the gaming environment, privacy can apply to the transmitted information as well as the identity of a player of the gaming machines.

In addition, information transmitted over the network bus should be authenticated. Authentication ensures that the content, integrity of the transmitted information, origin of the transmitted information, date of transmission, time of transmission and other attributes of the transmitted information have not been tampered with during transmission.

Additionally, entities transmitting information over the network bus should not be capable of repudiating the transmission. Cryptographic services that facilitate non-repudiation prevent a player and/or a casino from denying a previous action or commitment. The casino desires non-repudiation, especially, to enforce payment by a player that has wagered and lost. Conversely, the player desires non-repudiation to enforce payment by the casino when the player wins.

As a result of networking of the gaming machines, the ubiquity of the Internet, greater connectivity between networks, and the support for electronic commerce both inside and outside the casino, the casino desires secure communications over the network bus that provides privacy, authentication and non-repudiation. Therefore, a need exists to provide these services to support secure communication over the network bus between the gaming server and the gaming machines in a casino.

In addition, the casino may decide or desire to connect the gaming server and, hence, the network bus and all networked gaming machines, to an outside network. Networking the casino to an outside network may be advantageous for a gaming entity that owns several casinos in different locations. For example, the connection of each casino to a centralized computer would provide centralized accounting of financial information for all the casinos operated by the gaming entity.

If casinos are connected to outside networks, however, it is critical that communications originating within the casino (including gaming machines and the gaming server) remain secured against misuse or tampering by an unauthorized party after the information exits the physical protection of the casino. This desire for secured communications becomes particularly important when financial information is transmitted by the casino over the outside network. Consequently, a need exists for a secure communication link between the gaming server in a casino and an outside network.

In addition, the connection between gaming machines requires various transmission and/or data protocols. These protocols are typically created as standards in the industry. However, a game manufacturer would like to control the connection between the gaming machines such that only authorized personnel can connect the gaming machines. Therefore, a need exists for a technique to control the connection between the gaming machines such that only authorized personnel can properly connect the gaming machines.

Additionally, some casino players may prefer playing a specific gaming machine. However, the player may be in a remote location and unable to travel to the casino to play. In such instances, the casino can connect a gaming machine to an outside network so that the player can connect to the outside network via a remote computer and play, even though absent from the casino. In such instances, a need exits for a secure network that provides privacy, authentication and non-repudiation so that the player can play and both the player and casino can be confident in the knowledge that the transmitted information is secure and that the rules of the game will be upheld with integrity.

3. Solution to the Problem.

The present invention provides a method and apparatus that allows secure communication in a casino between networked gaming machines and a gaming server. With the present invention, privacy is ensured; communication is authenticated; and messages cannot be repudiated.

Additionally, the present invention discloses a method and apparatus that provides secure communications between the casino and an outside network. The present invention is especially advantageous if the gaming entity manages machines at multiple casinos in different locations and the gaming entity requires quick, yet secure retrieval of information over the outside network.

In addition, the present invention provides a method and apparatus for secure communications between each gaming machine. In this regard, this secure communication allows for the connection between the gaming machines to be controlled by the game manufacturer such that the gaming machines cannot be connected unless the cryptographic technique used to secure the communications between the gaming machines is known.

Lastly, the present invention provides secure communications between the casino and a remote player over an external network. The present invention is especially critical in ensuring that transmitted information between player and casino is kept confidential and indecipherable by unauthorized individuals intercepting the transmitted information.

SUMMARY OF THE INVENTION

[0013] Briefly stated, the present invention provides a system, apparatus, and method for implementing a game having a deterministic component and a non-deterministic component wherein a player uses the game through at least one player interface unit. Each player interface unit generates a player record indicating player-initiated events. A random number generator provides a series of pseudo-random numbers that are preferably statistically verified by integral verification algorithms and stored in a buffer. Preferably, the random number generator allows seed and key restoration automatically or manually upon power fault.

[0014] A rules library stores indexed rules for one or more games. An interface registry stores mapping records where the mapping records are used to associate the player-initiated events to pre-selected rules in the rules library. A control means is coupled to receive the output of the player interface unit, coupled to the interface registry, the rules library, and the random number generator. The control means processes the player record and returns an output record to the player interface unit where the output record is determined by executing the game's rules with reference to the pseudo-random numbers and predefined combinatorial algorithms for selecting sets of the pseudo-random numbers.

The present invention provides a casino gaming system having a plurality of gaming machines. In the Asymmetric case, a gaming server is provided that includes a plurality of long term keys from which it may generate keys used to communicate between gaming machines and also between the gaming machines and server. Prior to use, each of the keys is time stamped. The gaming server also includes a random number generator that is used to facilitate generation of the keys. The gaming server also includes an encryption algorithm.

A network bus is provided that interconnects the gaming machines and the gaming server. The network bus provides a communication link for transmitting information between the gaming machines and the gaming server. The gaming server uses the encryption algorithm to encrypt the keys and transmits the encrypted keys over the network bus to the gaming machine. Likewise, the gaming machines use the keys to encrypt information and transmit the encrypted information over the network bus. In one aspect, the encrypted information is transmitted via the network bus to another of the gaming machines. In another aspect, the encrypted information is transmitted via the network bus to the gaming server.

In another embodiment, the casino gaming system includes an outside network that is connected to the gaming server. A remote computer is also provided that connects to the outside network so that the encrypted information is transmitted over the network bus and the outside network to the remote computer. In one aspect, the outside network comprises the Internet.

In another embodiment of the present invention, the gaming server is a certificate authority server having a memory. In this aspect, the keys are public keys of asymmetric key pairs which are stored in the memory at the certificate authority server. In addition, the certificate authority server may generate and transmit the public keys over the network bus to the gaming machines, or the public/private key pairs may be generated by a third party and delivered to the certificate authority for authentication.

In a further embodiment of the present invention, a plurality of access switches are each connected to a different one of the gaming machines. The network bus is connected to the gaming server and each of the access switches. In this embodiment, an outside network is connected to the gaming server and the access switches provide a communication link between specific gaming machines and a remote computer over the outside network when the specific gaming machine is idle, so as to enable a remote player of the remote computer to play the specific gaming machine.

BRIEF DESCRIPTION OF THE DRAWING

[0015] FIG. 1 illustrates a simplified block diagram of the gaming engine in accordance with the present invention;

[0016] FIG. 2 illustrates a block diagram of the pseudo-random number subsystem in accordance with the present invention;

[0017] FIG. 3 illustrates the non-uniform distribution generator and combinatorial algorithm subsystems in accordance with the present invention;

[0018] FIG. 4 illustrates a main control circuit in accordance with the present invention;

[0019] FIG. 5 illustrates in block diagram form implementation of the rules library in accordance with the present invention;

[0020] FIG. 6 illustrates a flow chart of a game implementation using the apparatus shown in FIG. 1;

[0021] FIG. 7 illustrates a flow diagram for a second embodiment pseudo-random number distribution system;

[0022] FIG. 8 illustrates a multiple player networked implementation in accordance with the present invention; and

[0023] FIG. 9 illustrates in graphical form relationships between server speed, queue size, and customer wait times of an apparatus in accordance with the present invention.

Fig. 10 illustrates one embodiment of the casino gaming system of the present invention;

Fig. 11 is a flow chart showing a method for communicating information using a casino gaming system of the present invention;

Fig. 12 illustrates an embodiment of the casino gaming system of the present invention using a certificate authority server;

Fig. 13 illustrates another embodiment of the casino gaming system of the present invention; and

Fig. 14 is another embodiment of a method for communicating using the casino gaming system of the present invention.

DETAILED DESCRIPTION OF THE DRAWING

1. Overview

[0024] FIG. 1 illustrates, in simplified schematic form, a gaming apparatus in accordance with the present invention. The gaming apparatus in accordance with the present invention is also referred to as a "universal gaming engine" as it serves in some embodiments as a platform for implementing any number of games having deterministic and random components. In other embodiments, the universal gaming engine in accordance with the present invention provides a platform that supports multiple players across a network where each player preferably independently selects which game they play and independently controls progression of the game.

[0025] Although in the preferred embodiment all of the games discussed are implemented entirely electronically, it is a simple modification to alter the player interface to include mechanical switches, wheels, and the like. Even in mechanically implemented games electronic functions that are performed by the gaming engine in accordance with the present invention are required. Hence, these mechanical machines are greatly simplified using the gaming engine in accordance with the present invention.

[0026] Gaming engine 100 is illustrated schematically in FIG. 1, including major subsystems in the preferred embodiments. Each of the subsystems illustrated in FIG. 1 is described in greater detail below. FIG. 1, however, is useful in understanding the overall interconnections and functioning of the gaming engine in accordance with the present invention.

[0027] Gaming engine 100 performs several basic functions common to many electronically implemented casino games. The most basic of these functions includes interacting with the player to detect player initiated events, and to communicate the state of a game to the player. Gaming engine 100 must process the player initiated event by determining the appropriate rules of the game that must be executed and then executing the appropriate rules. Execution of the rules may require only simple calculation or retrieving information from memory in the case of deterministic rules, or may require access to pseudo-random values or subsets of pseudo-random values in the case of non-deterministic components.

[0028] Gaming engine 100 in accordance with the present invention uses a main control circuit 101 to control and perform basic functions. Main control circuit 101 is a hardware or software programmable microprocessor or microcontroller. Alternatively, main control circuit

101 can be implemented as an ASIC device with dedicated logic to perform the required control functions. Main control circuit 101 communicates with player interface unit 102 via interface bus 103. Player interface unit 102 is a machine having at least some form of display for communicating information to the player and some form of switch (i.e., buttons, levers, keyboard, coin slot, or the like) for communicating information from the player.

[0029] Player interface unit 102 generates a player record of information and transmits the player record over bus 103 to main control circuit 101. The player record of information contains information about the player initiated event as well as any data that may be associated with the particular event. For example, a player initiated event may be drawing two cards from a deck of cards. The player record will include information about the event (i.e., drawing cards), and data (i.e., two cards). The player record may include other information such as the state of the game that is being played. By "state of the game" it is meant at which stage in the rule defined progression of the game the game currently exists. State information may be maintained by gaming engine 100 or player interface unit 102, or both.

[0030] Main control circuit 101 responds to a player initiated event by referencing a public interface registry 107. Public interface registry 107 is essentially a lookup table implemented in volatile, semi-volatile, or non-volatile memory. Public interface registry 107 is desirably organized as an addressable memory where each address is associated with a mapping record. Main control circuit 101 uses the player event portion of the player record to address public interface registry 107 in a preferred embodiment. Public interface registry 107 then provides a selected mapping record to main control circuit 101. Main control circuit 101 uses the selected mapping record to address rules library 108.

[0031] Rules library 108 is essentially an addressable memory preferably allowing random access. Rules library 108 can be implemented in volatile, semi-volatile, or non-volatile memory of any convenient organizational structure. Rules library 108 responds to the address from main control circuit 101 by supplying one or more rules, which correspond to game rules, to main control circuit 101. The rules provided by rules library 101 are preferably executable instructions for main control circuit 101.

[0032] Main control circuit 101 processes the selected rules by selectively accessing random number circuit 104 and transform function algorithms 106. As set out herein before, completely deterministic rules may be executed entirely within main control circuit 101 by simple calculations or data transfer operations. Where the selected rule requires main control circuit 101 to access one or more pseudo-random numbers, random number circuit 104 is accessed. In the preferred embodiment random number circuit 104 provides a series of pseudo-random numbers of arbitrary length having uniform distribution as described in greater detail hereinafter.

[0033] Often times, however, a rule will require a non-uniform distribution of pseudo-random numbers, or some subset of a series of pseudo-random numbers. In this case, main control circuit 101 implements the selected rule by accessing transform function algorithms from block 106 in FIG. 1. The transform function algorithms transform the series of uniformly distributed pseudo-random numbers from random number circuit 104 by 1) transforming them into a non-uniform distribution, 2) using a given set of the uniformly distributed pseudo-random numbers to performing set selection permutations or 3) both.

[0034] In this manner, the basic functions of pseudo-random number generation, pseudo-random number transformation, and association of rules with player or player events are standardized and entirely contained in gaming engine 100. System operator interface 109 is used by the casino or game developer to communicate with uniform random number circuit 104 and main control circuit 101. This communication is desirable to initialize, program, and maintain main control circuit 101 and public interface registry 107, for example. System operator interface also enables an operator to initialize, monitor and change seed values and key values used by uniform random number circuit 104. Any convenient hardware may be used to implement system operator interface 109 including DIP switches, a smart terminal, personal computer, or a dedicated interface circuit.

[0035] To implement a game, a game programmer develops a series of rules for the game. The series of rules are stored as a volume in rules library 108. The game programmer will then register the new game in public interface registry 107 by storing the location of the volume of rules in an appropriate address in public interface registry 107. The game programmer does not need to program or develop the random number circuit or transform algorithms to implement a new game. Further, the player using player interface unit 102 can access any of the games stored in rules library 108. To certify a new game, a game regulatory authority need only review the rules in the rules library 108 to verify that they follow the established rules for a particular game. This verification can be easily done by reviewing high-level language code such as FORTRAN, C, or Basic.

[0036] While the present invention is described in terms of the preferred implementation of casino games it should be understood that any game which has a random component and progresses by following pre-defined rules can be implemented in gaming engine 100. Player interface unit 102 may be entirely electronic or combine electronic and mechanical components. Player interface unit may supply any amount and kind of information in addition to the basic functions set forth above to main control circuit 101. Player interface unit 102 may be located in the same physical machine as the remaining portions of gaming engine 100 or may be located at a great distance from gaming engine 100. These and other alternatives will be discussed in greater detail hereinafter.

2. Random Number Circuit

[0037] A preferred random number circuit 104 is shown in FIG. 2. Random number circuit 104 preferably includes random number generator circuit 201, verification algorithms 202, and buffer 203. Random number circuit 104 is controlled by random number control circuit 204 which is a microprocessor, microcontroller, or dedicated logic control circuit.

[0038] Random number generator circuit 201 provides a stream of uniformly distributed pseudo-random numbers on output 206. Alternatively, random number generator circuit 201 can provide parallel outputs on output 206. Also, more than one random number generator circuit 201 may be employed depending on the quantity of pseudo-random numbers demanded by the system.

[0039] Random number generator circuit 201 preferably supplies uniformly distributed pseudo-random numbers because a set of uniformly distributed numbers can be transformed easily by transform algorithms 106 into non-uniform distributions and combinatorial subsets. A preferred circuit for implementing random number generator circuit 201 is an ANSI X9.17 pseudo random number generator based upon a plurality of data encryption standard (DES) encryption circuits. Alternatively, random number generator circuit 201 may be implemented using the international data encryption algorithm (IDEA) encryption. Other random number generator circuits are known. When implementing other random number generator circuits 201, however, it should be appreciated that a high-quality, cryptographically strong pseudo-random number generator is preferable. A major advantage of the present invention is that the random number circuit 104 need be implemented only once to serve a plurality of games making it cost efficient to use relatively expensive circuitry to provide a high quality random numbered circuit 104.

[0040] Random number generator circuit 201 accepts as input one or more key values which are typically binary values having a fixed relatively large number of bits. For example, the ANSI X9.17 pseudo-random number generator uses 56-bit keys. Random generator circuit 201 also usually accepts a seed value, which is also another large bit binary value. Further, random number generator circuit 201 has a data input or clock input that accepts a continuously variable signal which is conveniently a clock representing on the clock or data input changes a new random number is output on line 206. Random number control circuit stores and provides the key values, seed value, and clock values to random number generator circuit 201.

[0041] A desirable feature in accordance with the present invention is that random number circuit 104 be able to boot up after a power fault (i.e., power is removed from the system) using the same seed values, key values, and clock value that existed before the power fault. This feature prevents a player or operator from continually resetting the system or gaining any advantage by removing power from gaming engine 100. One way of providing this functionality is to buffer the key values, seed values, and clock values in memory within random number control circuit 204 before they are provided to random number generator 201. After a power on default, circuit 104 can reboot autonomously using the values stored in buffers. Alternatively, new values can be provided via system operator interface 109 to ensure that the output after a power fault is in no way predictable based upon knowledge of output after a prior power fault.

[0042] In a preferred embodiment, random number generator circuit operates continuously to provide the series of random numbers on line 206 at the highest speed possible. By continuously, it is meant that random number generator circuit 201 operates at a rate that is not determined by the demand for random numbers by the rest of the system. Random number control circuit 204 provides key values, seed values, and data values to random number generator circuit 201 independently of any processing demands on main control circuit 101 (shown in FIG. 1). This arrangement ensures that random number circuit 104 operates at a high degree of efficiency and is not slowed down by computational demands placed on main control circuit 101.

In other words, the control circuit resources that implement random number control circuit 204 are independent of and usually implemented in a separate circuit from main control circuit 101.

[0043] Random number control circuit 204 accesses one or more verification algorithms 202 via connection 207. Verification algorithms 202 serve to verify that the raw random numbers on line 206 are statistically random to a predetermined level of certainty. Preferably, verification algorithms 202 include algorithms for testing independence, one-dimensional uniformity, and multi-dimensional uniformity. Algorithms for accomplishing these tests are well known. For example, independence of the pseudo random numbers can be performed by a Runs test. Uniformity can be verified by the Kolmorgorov-Smirnov or K-S test. Alternatively, a Chi-square test verifies uniformity. A serial test is an extension of the Chi-square test that can check multi-dimensional uniformity.

[0044] Random number control circuit 204 preferably receives and stores a set of raw random numbers from random number generator circuit 201. The set of raw random numbers can be of any size, for example 1000 numbers. Random number control circuit 204 then implements the verification algorithms either serially or in parallel to test independence and uniformity as described hereinbefore. It may be advantageous to use more than one physical circuit to implement random number control circuit 204 so that the verification algorithms may be executed in parallel on a given set of raw random numbers.

[0045] If a set of raw random numbers do not pass one of the verification tests the numbers are discarded or overwritten in memory so that they cannot be used by gaming engine 100. Only after a batch of numbers passes the battery of verification tests, are they passes via line 208 to verify random number buffer 203. Buffer 203 is preferably implemented as a first-in, first-out (FIFO) shift register of arbitrary size. For example, buffer 203 may hold several thousand or several million random numbers.

[0046] By integrating verification algorithms 202 in a random number circuit 104, gaming engine 100 in accordance with the present invention ensures that all of the pseudo-random numbers in buffer 203 are in fact statistically random. This overcomes a common problem in pseudo-random number circuits wherein the random numbers are long-term random, but experience short-term runs or trends. These short-term trends make prediction of both the player and casino odds difficult and may create an illusion of unfairness when none in fact exists. The verification algorithms 202 in accordance with the present invention largely eliminate these short-term trending problems and create a pool of random numbers in buffer 203 that are both statistically random and will appear to be random in the short run time period in which both the casino and players operate.

[0047] Buffer 203 makes the random numbers available continuously to main control circuit 101. Main control circuit 101 may access any quantity of the numbers in buffer 203 at a time. Buffer 203 also serves to provide a large quantity of random numbers at a rate higher than the peak generation rate of random number generator circuit 201. Although it is preferable that random number generator circuit 201 and verification algorithms 202 are processed so as to provide random numbers to buffer 203 at a higher rate than required by gaming engine 100, short-term bursts of random numbers can be provided by buffer 203 at a higher rate.

3. Transform Function Algorithms

[0048] Transform function algorithms 106 are accessed by main control circuit 101 as illustrated in FIG. 3. Examples of transform function algorithms 106 are a non-uniform distribution generator 301 and combinatorial algorithms 302. To execute some rules obtained from rules library 108, main control circuit 101 may be required to select one or more random values from a non-uniform distribution. Examples of non-uniform distributions are normal distribution, exponential distribution, gamma distribution, as well as geometric and hypergeometric distributions. All of these non-uniform distributions can be generated from the uniform distribution provided by random number circuit 104.

[0049] Rule implementations primarily require that main control circuit 101 access a series of pseudo-random numbers in the context of random set selection and permutations. This subset selection is performed by combinatorial algorithms 302. The combinatorial algorithms 302 operate on either the uniform number distribution provided directly by random number circuit 104 or the non-uniform distribution provided by non-uniform distribution generator 301. In this manner, a game of keno can be implemented by selecting a random 20 from a group of 80.

[0050] Another function of the transform algorithms 106 is to scale and center the series of random numbers. For example, a deck of cards includes 52 cards so that the set of random numbers must be scaled to range from 1 to 52. These and similar transform functions are well known.

[0051] An advantageous feature of the present invention is that these transform functions can be implemented a single time in a single piece of software or hardware and selectively accessed by any of the games in rules library 108. This allows a great variety of transform functions to be provided in a cost efficient and computationally efficient manner. The game designer need only provide rules in rules library 108 that access appropriate transform function algorithms 106 and need not be concerned with the details of how the transform function algorithms 106 are implemented. Similarly, a gaming regulatory authority can verify the correctness and fairness of transform algorithms a single time by providing extensive testing. Once the transform functions are verified, they need not be verified again for each game that is implemented in rules library 108. This independence between the rules programming and the non-deterministic programming result in highly standardized and reliable games while allowing the games designer greater flexibility to design a game in the rules library 108.

4. Main Control Circuit

[0052] A preferred embodiment of main control circuit 101 is shown in block diagram form in FIG. 4. Preferably, a micro-controller microprocessor 401 is provided to perform calculations, memory transactions, and data processing. Microprocessor 401 is coupled through bus 103 to player interface unit 102. Microprocessor 401 is also coupled to player number circuit 104, transform function algorithms 106, public interface registry 107, and rules library 108 through bi-directional communication lines 402.

[0053] In a typical configuration, main control circuit 101 will have a quantity of RAM/SRAM 403, a quantity of non-volatile memory 404, and ROM for storing an operating system and boot sequence. ROM 406 operates in a conventional manner and will not be described in greater detail hereinafter. Non-volatile memory 404 is an addressable, preferably random access memory used to store information that is desirably saved even if power is removed from main control circuit 101. For example, microprocessor 401 may calculate statistics regarding the type of games played, the rate of game play, the rate of number request, or information about the player from player interface unit 102. The statistics are preferably stored in a non-volatile memory 404 to maintain integrity of the information. Similarly, non-volatile memory 404 may be used to maintain the state of a game in progress on player interface unit 102 so that is power is removed, universal gaming engine 100 can restore player interface unit 102 to the state at which it existed prior to the power outage. This may be important in a casino operation where the casino could incur liability for stopping a game when the player believes a payoff is imminent.

[0054] RAM 403 serves as operating memory for temporary storage of rules access from rules library 108 or for storing the operating system for quick access. RAM 403 may also store groups of random numbers while they are being processed by the transform function algorithms as well as address data provided to and accepted from the public interface registry.

[0055] It should be understood that main control circuit 101 may be implemented in a variety of fashions using conventional circuitry. While some memory will almost surely be required, the memory may be implemented as RAM, SRAM, EPROM or EEPROM to meet the needs of a particular application. Similarly, the components of main control circuit 101 shown in FIG. 4 may be implemented as a single circuit or single integrated circuit or in multiple circuits or integrated circuits. Additional features may be added to implement additional functions in a conventional manner.

5. Rules Library

[0056] An exemplary embodiment of rules library 108 is illustrated in block diagram form in FIG. 5. Rules library 108 is preferably implemented as a plurality of volumes of rules where each volume is fixed in a rule EPROM 502-506. Any number of rule EPROM's can be supplied in rule library 108. Also, rule EPROM's 502 can be of various sizes. Rule EPROM's 502-506 may be replaced with equivalent memory circuits such as RAM, S RAM, or ROM. It is desirable from a gaming regulatory authority standpoint that rule EPROM's 502-506 cannot be altered once programmed so that the rules cannot be changed from the designed rules. This allows the gaming regulatory authority to verify the EPROM rules.

[0057] Address logic 501 provides address signals to select one of rule EPROM's 502-506. Additionally, address logic 501 serves to position a pointer to a specific rule within each rule EPROM 502-506. As set out herein before, which of rule EPROM's 502-506 is selected as determined by the current game being played as indicated by player interface unit 102 (shown in FIG. 1). The location of the pointer within a rule EPROM is addressed based upon the current state of the game and the particular user initiated event indicated by player interface unit 102. The information is conveyed from the user interface unit 102 in a player record that is mapped to rule library 108 by the information in public interface registry 107.

[0058] In practice, a game developer will program a series of rules that dictate the progression of a game in response to user or player initiated events. The rules will also dictate when random numbers are accessed and the type of random numbers which should be accessed (i.e., uniform or non-uniform distributions). Rules will also control payoffs, and place boundaries on the types of player events which will be accepted. The game developer will then burn these rules, once complete, into a rule EPROM, such a rule EPROM's 502-506 The rule EPROM can then be verified by a gaming regulatory authority, and once approved, be distributed to owners of gaming engines wishing to implement the newly developed game. In order to install the new game, the rule EPROM is installed in rules library 108 and registered in public interface registry 107. The registration process described hereinbefore provides gaming engine 100 the address information necessary to enable address logic 501 to access a particular rule in rules library 108 and provide that rule on output line 507 to main control circuit 101.

[0059] Although rules library 108 has been described in terms of a plurality of EPROM's 502-506 wherein each EPROM holds one volume of rules pertaining to a particular game, it should be apparent that many other configurations for rules library 108 are possible. Rules can be implemented in a single large memory or in a serial memory such as a tape or disk. Address logic 500 may be integrated in rules library 108, or may be integrated with main control circuit 101. Each game may be implemented in a single EPROM or may require several EPROM's depending on the particular needs of an application.

6. Method of Operation

[0060] FIG. 6 and FIG. 7 together illustrate in flow chart form a preferred method of operation of gaming engine 100 in accordance with the present invention. FIG. 6 details operation of a first embodiment single player gaming engine 100. When gaming engine 100 is started as indicated at 601 in FIG. 6, main control circuit 101 is initialized and goes through a boot-up sequence to bring it to an initial state. In this initial state it waits for user input at step 604. The player input or player record preferably indicates the game that is being played, the state of that game, and user initiated events and data that must be processed. Upon receipt of the player record, the public registry is addressed in step 606. The public registry returns a mapping record that matches the user record with a particular rule in the rules library in step 608.

[0061] One or more rules are accessed in step 608. Each of the one or more rules are processed in serial fashion in the embodiment illustrated in FIG. 6. One rule is processed in each pass through steps 610-622. A logical component of a first rule is processed in step 610, where the logical component includes processes of memory manipulations, calculations, and the like. In step 612, it is determined if the particular rule that was executed in step 610 requires pseudo-random numbers to process. If pseudo-random numbers are required, they are retrieved in step 700 which is illustrated in greater detail in reference to FIG. 7.

[0062] It is determined if the rule requires any transform algorithm in step 614. If a transform algorithm is required it is obtained in step 616. It should be understood that the transform algorithm may be permanently resident in the main control circuit 101 and so the step of obtaining 616 may be trivial. Once the necessary transfer algorithm is obtained, it is determined if the rule is completely processed in step 618. If not, flow returns to step 610 and the rule logic is executed until the rule is completely processed and a final result of the rule is determined. Once the rule is finished, control moves from step 618 to result accumulation step 620.

[0063] Each rule accessed in step 608 is processed in a similar manner by sequentially selecting each rule in step 626 until it is determined that all rules have been processed in step 622. Once all the rules are processed, the accumulated results are returned to the player in step 624. The results are of the rule are determined in steps 610, 612, and 614 by performing any transforms required on the random numbers, executing any deterministic components using conventional calculations and memory transactions.

7. Method for Random Number Generation

[0064] FIG. 7 illustrates a flow chart showing steps in filling random number request step 700 in FIG. 6. The process shown in FIG. 7 is initiated when request 614 is made. More accurately, many of the sub-processes shown in FIG. 7 are ongoing, but the processes for generating and supplying random numbers are also responsive to the request for random numbers 700.

[0065] Continuously ongoing processes include clock generation step 706, providing key value(s) step 710, and providing seed value(s) step 712. The clock signal generated in step 706 need not be a real time clock, nor does it have to provide a linearly increasing or decreasing output. It is sufficient that clock 706 output a continuously variable signal at a regular interval. As set out herein before, clock generation is preferably performed by random number control circuit 204 shown in FIG. 2.

[0066] In a preferred embodiment, a signal is generated by the occurrence of the player event. For example, the time of the player event is determined at step 704 and may be used as shown in FIG. 7. At step 708, the clock signal and the player event signal are combined to provide a continuously variable non-random signal. Where both the player event signal and the clock are digital, the combination can be realized as logical function such as AND, OR, XOR, NAND or the like. Also, the combination may be a concatenation or subtraction function. This feature of the present invention is optional, but adds a new degree of randomness.

[0067] At step 714, a series of raw random numbers is generated using the continuously provided key values, seed values, and variable signal. The raw random numbers are stored at step 716 to build a group large enough to be verified during step 718. Groups of raw random numbers that fail verification step 718 are discarded, while those that pass are stored at step 720 in buffer 203 shown in FIG. 2.

[0068] In accordance with a first embodiment, the verified random numbers are delivered in step 722, returning process flow to step 618 shown in FIG. 6. In an alternative embodiment shown in FIG. 7, request 614 is queued at step 728 using RAM 403 shown in FIG. 4. Request queuing 728 is implemented as a first in first out or "push up" register having N queue capacity. In one embodiment, N is between 2 and 10. Queuing step 728 stores each request and processes each request in turn. In this embodiment, delivery step 722 serves whatever request is provided during step 728. Once a request is delivered, the request queue is updated in step 724.

[0069] Although the request queue is optional, it increases efficiency of random number generation step 700. This is especially important in the networked multi-user embodiment shown in FIG. 8. FIG. 9 illustrates generally a relationship between server speed, queue size, and the average number of customers, or requests for pseudo-random numbers, are waiting in the system. FIG. 9 is derive by modeling gaming engine 800 (shown in FIG. 8) as an M/M/1 queue to produce parameters for expected wait times in the system. FIG. 9 assumes that requests for pseudo-random numbers are made according to a Poisson process. This means that the times between successive arrivals are independent exponential random variables.

[0070] Upon arrival, a customer either immediately goes into service if the server is free, or joins queue 728 if the server is busy. When step 722 finishes obtaining the requested subset, the request is returned to the game and leaves the system. The next request, if any, is serviced. The times required to form the requested random subsets are assumed to be independent exponential random variables also. With these assumptions, request queue 728 can be viewed as an M/M/1 queue. The first two M's indicate that both the interarrival times as well as the service times for requests are exponential random variables. The "1" indicates there is just one server.

[0071] Server speed is largely determined by the hardware chosen to implement the present invention, and can be easily varied by those of skill in the art to meet the needs of a particular application. As is apparent in FIG. 9, higher server speeds result in fewer waiting customers. From the lower portion of FIG. 9, is apparent that if the queue size is reduced to zero (i.e., no request queue), the average wait time climbs even with very fast servers. Hence, to minimize wait time, a request queue is desirable.

[0072] It should be understood that the process steps shown in FIG. 7 may be carried out in any convenient order unless expressly specified above. Process steps may be carried out in serial or parallel depending on the particular capabilities of main control circuit 101 shown in FIG. 1. For example, where control circuit 101 is multi-tasking or capable of parallel processing, several process steps may be executed at once. Also, process steps may be added to those shown in FIG. 7 to implement additional functions without departing from the inventive features of the present invention.

8. Network Embodiment

[0073] FIG. 8 illustrates in block diagram for a network embodiment in accordance with the present invention. Basic components of gaming engine 800 are similar to gaming engine 100 including random number circuit 804, transform algorithms 806, public interface registry 807, and rules library 808. Main control circuit 801 includes all of the functions described herein before in reference to main control circuit 101 but also includes function for supporting network interface circuit 812. Data bus 812 couples main control circuit 801 to network interface circuit 812.

[0074] The network embodiment shown in FIG. 8 serves a plurality of player interface units 802a-801e. This additional functionality is provided in part by network interface circuit 812 and network I/O circuits 812a-812e. Network interface circuit 812 and network I/0 circuits 812a-812e can be conventional network circuits used for 10baseT, ethernet, Appletalk, or other known computer network systems. In selecting the network circuits, it is important that the data throughput is adequate to meet the needs of a particular system.

[0075] Network interface circuit 812 communicates a plurality of player records of information to main control circuit 801. Main control circuit may be a conventional processing circuit that serially processes each of the player records in a manner similar to main control circuit 101. Preferably, main control circuit 801 includes multitasking or parallel processing capabilities allowing it to process the plurality of player records simultaneously.

[0076] Simultaneous processing requires that main control circuit 801 access a plurality of rules from rules library 808, each of which may require main control unit 801 to request a set of pseudo-random numbers from random number circuit 804. In a preferred embodiment, the multiple requests for pseudo-random numbers are stored in a request queue implemented in memory of main control circuit 801. The request queue is preferably able to store more than one request. A suitable request queue can store ten requests. Random number circuit 804 treats each request from the request queue of main control circuit 801 in a manner similar to the requests from main control circuit 101 described herein before. The combination of the request queue with the buffer of random number circuit 804 allows gaming engine 800 to service requests corresponding to player initiated events very efficiently. A request queue holding even two or three requests can reduce the probability of any player waiting for delivery of a set of pseudo-random numbers significantly.

[0077] The request queue can be implemented by configuring a portion of the RAM available to main control circuit 801 as a first-in first-out register or push up stack. Each request for a set of random numbers is initially placed at the bottom of the request queue and sequentially raised in the request queue until the request is filled. This operation is described herein before with respect to FIG. 7.

In Fig. 10, a highly simplified gaming system 1100 includes a gaming server 1110 that is connected to a plurality of gaming machines 1120-1124 via network bus 1130. The gaming server 1110 can comprise, for example, a micro-computer or a network server. The connection of the gaming server 1110 to network bus 1130 can comprise, for example, a hard-wired communication link connection or a wireless communication link connection. The network bus 1130 also connects to the plurality of gaming machines 1120-1124 that are located in a casino. In one embodiment, the gaming machines 1120-1124 can comprise conventional stand-alone gaming machines that are networked to the gaming server 1110 via the network bus 1130. The gaming machines 1120-1124 can also allow play of various conventional casino games such as, but not limited to, slots, poker, blackjack, etc.

In one embodiment, the casino gaming system 1100 can also includes an outside network 1140, such as, for example, the Internet, a Local Access Network (LAN) or a Wide Area Network (WAN). At least one remote computer 1150 is connected to the outside network 1140. In one embodiment, the connection of the remote computer 1150 to the outside network 1140 also enables the remote computer 1150 to connect to the gaming server 1110, and hence, the network bus 1130 and the gaming machines 1120-1124. In this embodiment, a remote player of the remote computer 150 can play a specific one of the gaming machines 1120-1124 on the network bus 1130 through the connection to the outside network 1140. As such, a remote player can play a specific one of the gaming machines 1120-1124 via an outside network 1140 without having to be physically in the casino.

The connection between the network bus 1130 and the gaming server 1110 is conventionally known in the art, and the connection can include other equipment (not shown) such as, for example a router. The connection between the gaming server 1110 and the outside network 1140 is also known in the art, and the connection can include various security features, such as, for example, a firewall. The connection between the remote computer 1150 and the outside network 1140 can include, for example, a hardwire connection, a wireless connection or a modem connection. It should be appreciated that the present invention is not limited to the manner in which the components are connected, since such connection of the components is known in the art.

In the gaming system 1100 of the present invention, information that is transmitted over the network bus 1130 and the outside network 1140 must be secure, especially with regard to financial information, such as, for example, player credit card information, player wagering information and casino pay-out information. To ensure a secure transmission of information over the network bus 1130 and the outside network 1140, the information is encrypted using various cryptography techniques. Key cryptography and certificate authority techniques are described below with regard to secure encrypted information transmission in a casino gaming system 1100.

II. Key Cryptography

A. Casino Gaming System using Keys

In Fig. 10, the casino gaming system 1100 includes a network 1130 that interconnects gaming machines 1120-1124 and gaming server 1110. The network bus 1130 provides a communication link for transmitting information between the gaming machines 1120-1124, themselves, and between the gaming machines 1120-1124 and the gaming server 1110. It should be noted that the computational capabilities of the gaming server 1110 should generally exceed those of the gaming machines 1120-1124, at least with respect to cryptographic operations. In this regard, many computer systems have architecture and/or use compilers that physically limit the bit length of an integer, such as, for example, 32 bit length. However, key cryptography requires the use of very large integers having a bit length such as, for example, 64 or 256 bits. To enable these computer systems to arithmetically manipulate these integers, cryptographic primitives are required. Cryptographic primitives include algorithms that process large integers during various arithmetic processes. It should also be noted that these cryptographic primitives can be any algorithm that allows processing of large bit length integers by these bit-limited computer systems.

However, these primitives should be able to support Rivest, Shamir, and Adleman (RSA), EI Gamal and other known key cryptographic algorithms. It should also be appreciated that the present invention is not limited by the algorithms and/or cryptography used to manipulate these large bit length integers, and the present invention encompasses any technique known and practiced in the art.

The gaming server 1110 also includes keys 1160. For example, the keys 1160 can comprise, as will be described later, symmetric keys, asymmetric keys or session keys. The keys 1160 include a time stamp 1165 that indicates a period of time for which each of the keys 1160 is valid. The time stamp 1165 also ensures that the keys 1160 are changed on a periodic basis to provide a more secure communication link.

The gaming server 1110 also includes a random number generator 1170 that is used by the gaming server 1110 to generate the keys 1160. The random number generator 1170 can comprise a pseudo-random number generator and/or a random number generator that has been approved by a governmental regulation agency. The generation of the keys 1160 by using the random number generator 1170 is known in the art and the present invention should not be limited to anyone technique for generating the keys 1160. It should also be appreciated that in another embodiment the random number generator 1170 is optional. In this embodiment, the gaming server 1110 receives the keys 1160 from another device (not shown) connected to the network bus 1130.

It should also be appreciated that the gaming server 1110 can also include an encryption algorithm 1180. The gaming server 1110 uses the encryption algorithm 1180 to encrypt information or data before it is transmitted over the network bus 1130. The encrypted information is decrypted before it is used. The encryption algorithm 1180 can comprise, for example, a symmetric key or one of an asymmetric key pair as will be explained herein below.

In one embodiment, the gaming server 1110 transmits at least one of the keys 1160 over the network bus 1130 to a gaming machine 1120. It should be appreciated that the gaming server 1110 can transmit one of the keys 1160 to anyone of the gaming machines 1120-1124 on the network bus 1130. However, for ease of description, this discussion will focus on transmission to gaming machine 1120.

In this embodiment, the gaming machine 1120 uses the keys 1160 to encrypt information, such as, player credit card information, player identification information, wagering information and casino payout information. This encrypted information is transmitted over the network bus 1130.

In one aspect, the encrypted information is transmitted over the network bus 1130 to the gaming server 1110. In another aspect, the encrypted information is transmitted over the network bus 1130 from gaming machine 1120 to another of the gaming machines 1122-1124. At the other gaming machine 1122-1124, the encrypted information is decrypted based on the type of key 1160 used as will be described herein below.

In another embodiment, the casino gaming system 1100 includes an outside network 1140 that is connected to the gaming server 1110. The outside network 1140 is connected to a remote computer 1150. The outside network 1140 comprises, for example, the Internet, a local access network (LAN) or a wide area network (WAN). In this embodiment, the gaming server 1110 includes various known security mechanisms (not shown), such as, a firewall.

In this embodiment, the gaming server 1110 transmits the key 1160 to gaming machine 1120. The gaming machine 1120 encrypts information using the key 1160 and transmits the encrypted information over the network bus 1130. In one aspect of the present invention, the encrypted information is transmitted to the gaming server 1110. In another aspect of the present invention, the encrypted information is transmitted by the gaming machine 1120 to another of the gaming machines 1122-1124 on the network bus 1130. In even another aspect, the encrypted information is transmitted to the outside network 1140 and, ultimately, to the remote computer 1150. Once the encrypted information has been received, it is decrypted based on the type of key 1160 used, as will be described herein below. The information is then processed as required.

1. Symmetric Keys

As explained above, the keys 1160, in one embodiment, comprise symmetric keys. Symmetric keys, also termed private keys, use a unique key to encrypt and exchange information between two parties. In this embodiment, gaming machine 1120 (the sender) and gaming server 1110 (the recipient) share a symmetric key *k* which is secret. In this embodiment, the gaming machine 1120 encrypts information *m* before transmitting it over the network bus 1130 to the gaming server 1110. If symmetric encryption algorithm *E* and symmetric key *k* are used, the encryption of *m* by *E* under *k* is denoted $c = E_k(m)$ where *c* represents the cipher-text associated with information *m*. Therefore, gaming machine 1120 transmits cipher-text *c* over the network bus 1130 to the gaming server 1110. At the gaming server 1110, the cipher-text *c* is decrypted using the symmetric key *k*. The gaming server 1110 applies the decryption algorithm $m = E_k^{-1}(c)$ to decrypt cipher-text *c* and obtain information *m*.

In addition, the symmetric key *k* can also be a session key. A session key is used for a specific exchange of a message *m* between two parties, such as, two gaming machines 1120 and 1122 or between a gaming machine 1120 and the gaming server 1110. In this embodiment, gaming machine 1120 desires to communicate with gaming machine 1122. The gaming server 1110 contains a symmetric encryption function ,E, that allows the encryption of a *session key, k,* that will be sent by the gaming server 1110 in an encrypted format to gaming machines 1120 and 1122. In this embodiment, $E_{k_i}(m)$ represents the encryption of message *m* under encryption algorithm *E* using key k_i, and $E_{k_i}^{-1}(m)$ represents the decryption of message *m* under encryption algorithm *E* using key k_i. In order to allow the communication between gaming machines 1120 and 1122, the gaming server 1110 generates a new unique session key *k*, and the gaming server 1110 sends $E_{k1}(k)$ *to* gaming machine 120 and $E_{k2}(k)$ to gaming machine 1122. The gaming machines 1120 and 1122 each can recover the session key *k* by forming $k = E_{k1}^{-1}(E_{k_2}(k)) = E_{k_{21}}^{-1}(E_{k_2}(k))$. Using the session key *k*, gaming machine 1120 can communicate message m to gaming machine 1122 by sending $E_k(m)$ *to* gaming machine 1122, gaming machine 1122 can form $m = E_k^{-1}(E_k(m))$ *to* recover the message. It should be appreciated that this technique can be used with communications between any device connected to the network bus 1130 and should not be limited to communications between only gaming machines 1120 and 1122. In addition, in one embodiment, the gaming server 1130 generates the session key k using the long term asymmetric key 1160 as a seed to random number generator 1170. In another embodiment, the gaming server can use anyone way function that is non-invertable to generate the session key k. However, it should be appreciated that the present invention can use any technique known in the art to generate the session key k, and the present invention should not be limited to only those disclosed. It should be noted that the cipher-text c is described as being transmitted only from the gaming machine 1120 to the gaming server 1110. However, it should be understood that the cipher-text c can be transmitted from the casino gaming server 1110 to the gaming machine 1120 using the same symmetric key 1160. Moreover, it should be appreciated that cipher-text c can be transmitted from anyone of the gaming machines 1120-1124 to the gaming server 1110 or vise versa using the symmetric key 1160. In addition, the cipher-text c can be transmitted from the gaming machines 1120-1124 or the gaming server 1110 to the outside network 1140 and the remote computer 1150 (or vise versa) using the symmetric key 1160 as described above. It should further be appreciated that the encryption algorithm 1180 used by the gaming server 1110 to encrypt and transmit the keys 1160 to the gaming machines 1120-1124 can comprise a symmetric key 1160, and the key 1160 can be encrypted and/or decrypted as described above with reference to information m. In a preferred embodiment, the symmetric key 1160 uses the Data Encryption Standard (DES) or one of the variants of DES such as triple-DES, DES-X or Advanced Encryption Standard (AES).

2. Asymmetric Keys

As mentioned above, the keys 1160 can comprise asymmetric keys. Asymmetric keys, also termed public keys, use two different keys in a transaction. The asymmetric key pair consists of a public and a private key. The public key is made available to all devices on the network bus 1130 and the outside network 1140 while the private key is kept secret. The essential feature of a public key cryptographic system is that knowledge of a public key does not provide computational information about the private key.

In this embodiment, the asymmetric key pair 1160 is represented by *(u,r)* where *u* represents the public key and *r* represents the private key. The gaming machine 1120 acquires the public key *u* of the gaming server 1110 from the gaming server 1110 or another device (not shown) connected to the network bus 1130 or the outside network 1140. The gaming machine 1120 encrypts information *m* using public key algorithm E_u. As a result, the cipher-text *c* is $c = E_u (m)$. The cipher-text *c* is transmitted to the gaming server 1110 over the network bus 1130. The private key algorithm E_r^{-1} is used by the gaming server 1110 to decrypt the cipher-text *c* and therefore obtain the information $m = E_r^{-1}(E_u(m))$. In this embodiment, it should be appreciated that each of the gaming machines 1120-1124, the gaming server 1110 and the remote computer 1150 have a unique asymmetric key pair *(u,r)*. The public key *u* is provided to the sending party and only the private key *r* can decrypt information encrypted by the public key *u*. It should also be appreciated that the asymmetric key technique can be used by any device connected to the network bus 1130 or the outside network 1140 so long as the appropriate public key *u* is used to encrypt the information *m* and the cipher-text *c* is sent to the device having the corresponding private key *r*.

In addition, it should also be appreciated that the encryption algorithm 1180 can comprise the public key *u* of the asymmetric key pair *(u,r)*. The gaming server 1110 encrypts the key 1160 using the public key *u* and transmits the encrypted key 1160 to the appropriate gaming machine 1120-1124 or remote computer 1150 having the corresponding private key *r*. In a preferred embodiment of the present invention, the asymmetric keys 1160 comprise Rivest, Shamir, and Adleman (RSA) and EI Gamal asymmetric algorithms.

3. Digital Signatures

In another embodiment, the keys 1160 can comprise a digital signature. A digital signature can be constructed by reversing the asymmetric key technique described above. In this embodiment, the gaming machine 1120 uses the private key algorithm E_r^{-1} to encrypt the information *m* where the cipher-text is $c = E_r^{-1}(m)$. The cipher-text *c* is transmitted to the gaming server 1110 where the cipher-text *c* is decrypted to obtain information *m* by applying the public key algorithm $m = Eu(E_r^{-1}(m))$. Since the private key algorithm E_r^{-1} is only known by the gaming machine 1120, the gaming server 1110 can be particularly certain that the information m was sent by the gaming machine 1120 because only the public key algorithm E_u is able to decrypt cipher-text c that has been encrypted using the private key algorithm E_r^{-1}.

As shown above, the digital signature is a variation of the asymmetric key technique described above and can be fully implemented using asymmetric keys. The digital signature provides an extra security feature that allows the receiving party to verify the sending party. This technique is particularly useful in the casino gaming system 100 when financial information, such as, credit card information, is being transmitted over the network bus 1130.

It should be appreciated that the digital signature has been disclosed with reference to the gaming machine 1120 and the gaming server 1110 but should not be limited as such. The digital signature can be used by all devices connected to the network bus 1130 and/or the outside network 1140. In addition, the encryption algorithm 1180 used by the gaming server 1110 to encrypt and transmit keys 1160 over the network bus 1130 can comprise a digital signature.

B. Method For Using Keys

As shown in Fig. 11, the present invention includes a method for communicating information using a casino gaming system 100 having gaming machines 1120-1124 and a gaming server 1110. The method includes establishing a first communication link (network bus 1130 in Fig. 10) between the gaming machines 1120-1124 and the gaming server 1110 (step 1210). A second communication link (outside network 140 in Fig. 10) is established between the gaming server 1110 and the remote computer 1150 (step 1220). It should be appreciated that the outside network 1140 can comprise the Internet, a local access network (LAN) or a wide area network (WAN).

The gaming server 1110 includes keys 1160. In one embodiment, the gaming server 1110 includes a random number generator 1170 that randomly generates the keys 1160 (step 1230). The gaming server 1110 can also include an encryption algorithm 1180 that is used to encrypt the keys 1160 at the gaming server 1110 (step 1240). It should be appreciated that the keys 1160 and the encryption algorithm 1180 can comprise symmetric keys or asymmetric keys that function as described herein above.

The key 1160 is transmitted from the gaming server 1110 to, in one embodiment, a gaming machine 1120 (step 1250). It should be appreciated that the gaming server 1110 can transmit the key 1160 to any other device connected to the network bus 1130 or the outside network 1140. The key 1160 is used by the gaming server 1110 to encrypt information sent from the gaming machine 1120 (step 1260). The encrypted information is transmitted over the first communication link (network bus 1130) and/or the second communication link (outside network 1140) (step 1270). It should be appreciated that the encrypted information can be transmitted to another of the gaming machine 1122-1124, the gaming server 1110 or the remote computer 1150. Once the encrypted information is received, it is decrypted by the receiving device (such as, for example, gaming server 1110) using a technique based on the type of key 1160 used as described herein above (step 1280).

It should be appreciated that the method described with reference to gaming machine 1120 and gaming server 1110 is only for ease of description and should not be interpreted as being limited as such. It should be appreciated that the above described method can be used by any device connected to the network bus 1130 and/or the outside network 1140.

III. Certificate Authority

In general, as shown in Fig. 12, a certificate authority server 1300 guarantees the identity of a device connected to the network bus 1130 or connected to the outside network 1140. The certificate authority 1300 guarantees the identity by granting a unique public key 1315 to each of the devices (as shown in Fig. 12, such as, gaming machines 1120-1124, gaming servers 1330-1332 and certificate servers 1340-1342) connected to the network bus 1130. The certificate authority server 1300 can also grant a unique public key to certain devices (such as remote computer 1150) that are connected to the outside network 1140. As noted above, there can be other certificate authority servers 1340 and 1342 connected to the network 1130. All the certificate authority servers 1300, 1340 and 1342 can be connected in a hierarchical configuration which is known in the art. In addition, there may be gaming servers 1330-1332 that do not have the ability to guarantee the identity of a device connected to the network bus 1130. However, these gaming servers 1330-1332 have the ability to perform other operations on the network bus 1130, as described above with reference to Fig. 10.

A. Casino Gaming System using a Certificate Authority

As shown in Fig. 12, another embodiment of the casino gaming system 1100 includes a certificate authority server 1300 that is used for communicating information using asymmetric key pairs including a private key and a public key. In this embodiment, a network bus 1130 interconnects the certificate authority 1300 and the gaming machine 1120-1124. The network bus 1130 can also be connected to other certificate authority servers 1340-1342 and gaming servers 1330-1332. The certificate authority server 1300 includes a memory 1310 that stores public keys 1315. The public keys 1315 can also include a time stamp (not shown) that indicates a time period that the asymmetric key pair is used. The certificate authority server 1300 also includes a random number generator 1320 that is capable of generating the asymmetric key pairs of the present invention.

The certificate authority server 1300 is also connected to an outside network 1140 and a remote computer 1150 is connected to the outside network 1140. The outside network 1140 can comprise the Internet, a local access network (LAN) or a wide area network (WAN). In another embodiment, it should be appreciated that the outside network 1140 can connect to a gaming server 1330-1332 or another certificate authority server 1340-1342. The certificate authority server 1300 can include other security mechanisms (not shown) to facilitate connection to the outside network 1140, such as, for example, a firewall. The remote computer 1150 can connect to the outside network 1140 via, a hard wired connection, a wireless connection or a modem connection.

For ease of discussion, the certificate authority server 1300 will be described with regard to transmissions to and from gaming machine 1120 and gaming server 1330. However, it should be appreciated that the certificate authority server 1300 can transmit to any device on the network bus 1130 and/or the outside network 1140, and these devices can communicate using the same techniques as previously described with regard to the gaming machine 1120 and the gaming server 1110 (in Fig. 10).

In the present embodiment, when the gaming machine 1120 desires to communicate with the gaming server 1330, the gaming machine 1120 requests a public key 1315 from the certificate authority server 1300. The certificate authority server1300 transmits a public key 1315 to the gaming machine 1120. The public key 1315 is used by the gaming machine 1120 to communicate with the gaming server 1330 connected to the network bus 1130. Prior to transmission of the public key 1315, the certificate authority server 1300 has verified the identity of the gaming server 1330 and granted a unique asymmetric key pair to the gaming server 1330. The verification is accomplished using various techniques known in the art. As a result of this verification, the certificate authority server 1300 can guarantee the identity of the gaming server 1330 and the validity of the public key 1315 that is to be used by the gaming machine 1120 to communicate with the gaming server 1330.

In addition to transmitting the public key 1315, the certificate authority server 1300 signs the public key 1315. The signing of the public key 1315 uses an encryption algorithm that is similar to the symmetric and asymmetric keys, such as, a digital signature, as described above. Once the gaming machine 1120 receives the signed public key 1315, the public key 1315 is validated using, as described above, symmetric or asymmetric key techniques. The gaming machine 1120 uses the public key 1315 to encrypt information and transmits that information over the network bus 1130 to the gaming server 1330.

As explained above, the gaming machine 1120 can communicate with any other device connected to the network bus 1130 and/or the outside network 1140. However, these other devices must also be verified by the certificate authority server 1300. As a result, the gaming machine 1120 receives the appropriate public key 1315 and transmits encrypted information to the appropriate device, such as, for example, other gaming machines 1122-1124, gaming severs 1330-1332, certificate authority servers 1300, 1340-1342 and remote computer 1150. In a preferred embodiment, the certificate authority server 1330 meets the X.509 (ISO/IEC 9594-8) standard.

IV. Remote Access

As shown in Fig. 13, another embodiment of the casino gaming system 1100 includes switches 1420, 1422 and 1424 that enable a remote player using a remote computer 1150 to connect to and play a specific gaming machine 1120-1124 that is located in a casino.

A. Remote Access Casino Gaming System

In this embodiment, shown in Fig. 13, a network bus 1130 interconnects a gaming server 1110 and switches 1420, 1422 and 1424. A certificate authority server 1300 is also connected to the network bus 1130. The certificate authority server 1300 provides public keys 1315 used for encrypting communications, as described above. The switches 1420, 1422 and 1424 are connected to gaming machine 1120, 1122 and 1124, respectively. In a preferred embodiment, the gaming machines 1120, 1122 and 1124 are located in a casino. However, the physical location of the gaming machines 1120, 1122 and 1124 should not be interpreted as limiting the present invention. The gaming server 1110 is connected to an outside network 1140 and a remote computer 1150 is connected to the outside network 1140.

In another embodiment, the outside network 1140 can connect to the certificate authority server 1300. The gaming server 1110 can have various security features to facilitate connection to the outside network 1140, such as, for example, a firewall. The outside network 1140 can comprise the Internet, a local access network (LAN) or a wide area network (WAN). The remote computer 1150 can be connected to the outside network 1140 via a hard wired connection, a wireless connection or a modem connection.

The present invention allows a remote player using a remote computer 1150 to connect to and playa specific gaming machine 1120-1124 in a casino. For ease of description, the remote computer 1150 will be described as connecting to gaming machine 1120. However, it should be noted that the present invention encompasses the remote computer 1150 connecting to any of the gaming machine 1122-1124 that are connected to the outside network 1140. As such, the remote computer 1150 connects to the outside network 140 which is connected to the gaming server 1110. The remote computer 1150 can be located in the casino, or the remote computer 1150 can be located remotely from the casino, such as, but not limited to, a hotel connected to the casino.

To play the gaming machine 1120, the remote computer 1150 makes a request to the gaming server 1110 to gain access to gaming machine 1120. The request made by the remote computer 1150 can include entering identification information that uniquely identifies the remote player of the remote computer 1150. The identification information can comprise a password, credit card information, etc.

The gaming server 1110 compares the identification information with a database. The database can include a listing of all passwords, a credit check of the credit card information or casino-specific credit information. If the identification information matches one of the entries in the database, the remote computer 150 is given access to the gaming machine 1120 through switch 1420.

It should be appreciated that, in another embodiment, the switch 1420 disconnects the gaming machine 1120 from access by the remote computer 1150 when the gaming machine 1120 is being used in the casino. The disconnection of the gaming machine 1120 can be initiated by a casino player in the casino. In this embodiment, if a casino player in the casino does not want a remote player connecting to the gaming machine 1120, the casino player can activate switch 1420 to prevent a remote player from accessing the gaming machine 1120.

In addition, governmental regulation may require that only one person at a time can playa gaming machine 1120 in the casino. In this case, the remote computer 1150 receives a gaming machine unavailable signal when the gaming machine 1120 is occupied and/or not idle, and the remote computer 1150 is asked to choose another gaming machine 1122-1124. Conversely, if a remote computer 1150 is accessing the gaming machine 1120, a casino player cannot play the accessed gaming machine 1120. In the casino, this disconnection is indicated by a light (not shown) or other indicators that verify that the gaming machine 1120 is unavailable.

Once the remote computer 1150 gains access to the gaming machine 1120, the remote player can play the gaming machine 1120. In one embodiment during play of the gaming machine 1120, the remote player views a digital representation of the game being played on the gaming machine 1120. The remote player can view and interact with the gaming machine 1120 via other mechanisms that are known in the art.

The present invention should not be interpreted as being limited to the manner in which the remote player views and interacts with the play of the gaming machine 1120. Furthermore, if the gaming machine 1120 breaks down or malfunctions during play, the gaming machine 1120 sends a signal to the remote computer 1150 indicating that the gaming machine 1120 is no longer available and the remote player is asked to play another game and is credited any winnings from the gaming machine 1120.

In addition, the communication between the remote computer 1150 and the gaming machine 1120 can be encrypted using symmetric or asymmetric keys as described herein above. The gaming server 1110 or the gaming machine 1120 can document information with regard to the wagering during .remote play of the gaming machine 1120. Such information can include identification information about the remote player, amounts wagered, the time the remote player plays the gaming machine 1120 and the location from which the remote player is playing the gaming machine 1120.

B. Method Remotely Accessing Casino Gaming System

As shown in Fig. 14, a method is provided that allows a remote player to access and play a specific gaming machine 1120-1124 from a remote location. In this method, a request is received from an outside network 1140 to access and play a gaming machine 1120-1124 (step 1510). The request from the outside network 1140 may be initiated by the input of identification information. The identification information can comprise a password, credit card information, etc. The gaming server 1110 compares the identification information with a database.

The database can comprise a listing of all passwords, a credit check of the credit card information or casino-specific credit information. If the identification information matches one of the entries in the database, the remote computer 1150 is given access to the gaming machine 1120 through switch 1420. It should be appreciated that, in another embodiment, the switch 1420 disconnects the gaming machine 1120 from access by the remote computer 1150 when the gaming machine 1120 is being played in the casino. It should further be appreciated that the present invention is not limited to the type of request that is made by the remote computer 1150 for access to the gaming machine 1120.

Based on the request, a secured communication link is provided between the outside network 1140 and the gaming machine 1120-1124 (step 1530). In one embodiment, the secured communication link is only provided if the gaming machine 1120-1124 is idle and/or not being played by another player (step 1520). In this embodiment, if the gaming machine 1120 is not idle, a gaming machine unavailable message is provided to the outside network 1140 (step 1540). Additionally, the remote player can be asked to choose another of the gaming machines 1122-1124.

Once the outside network 1140 accesses a gaming machine 1120-1124, information can be documented (step 1550). The information can include identification information about the remote player, amounts wagered, the time the remote player plays the gaming machine 1120 and the location from which the remote player is playing the gaming machine 1120. When the remote player begins to play, the player views a digital representation of the gaming machine 1120.

The foregoing discussion of the invention and as presented in Exhibit A (incorporated herein by reference) has been presented for purposes of illustration and description. Further, the description is not intended to limit the invention to the form disclosed herein. Consequently, variation and modification commensurate with the above teachings, within the skill and knowledge of the relevant art, are within the scope of the present invention.

The embodiment described herein and above is further intended to explain the best mode presently known of practicing the invention and to enable others skilled in the art to utilize the invention as such, or in other embodiments, and with the various modifications required by their particular application or uses of the invention. It is intended that the appended claims be construed to include alternate embodiments to the extent permitted by the prior art.

[0078] By now it should be appreciated that an apparatus, method, and system for gaming is provided with greatly improved efficiency and quality over existing gaming methods and systems. The universal gaming engine in accordance with the present invention is a gaming apparatus providing a consistent game development platform satisfying the needs of gaming authorities, house, player, and game developer. The gaming engine in accordance with the present invention separates the problems of developing game rules from the difficulty of producing chance events to support those rules. By including basic functions shared by a number of games, hardware costs are greatly reduced as new games can be implemented merely by providing a new set of rules in the rules library and the basic hardware operating the game remains unchanged. It is to be expressly understood that the claimed invention is not to be limited to the description of the preferred embodiments but encompasses other modifications and alterations within the scope and spirit of the inventive concept.

3109138
111306